Paper Summaries

Arka Sadhu

August 1, 2017

Contents

1 Unsupervised Co-segmentation for Indefinite Number of Common Foreground Objects

[1]

1.1 Abstract

- Co-segmentation addresses the problem of simultaneously extracting the common targets appeared in Multiple images.
- Keywords : Co-segmentation, multi-object discovery, adaptive feature, loopy belief propagation

1.2 Introduction

The paper extends the previous proposal Selection based Co-segmentation[PSCS] methods with the 3 major contributions :

- Key problem in [PSCS] is mining consistent information shared by the common targets. May require manual selection of features, or feature learning performed beforehand. Here : (simple and effective) self-adaptive feature selection strategy is introduced.
- Many assume each image contains a single common target and fail for multiple common targets images to extract all targets. Here proposal selection based Unsupervised Co-segmentation [PSUCS] is introduced.
- For multiple common targets, multi-class co-segmentation approaches do not do so well because of significant appearance variance and the inconsistent number of common targets, also some combinational common targets are usually split into multiple pieces. Here : an adaptive strategy that can handle Indefinite number of common targets involved cases, where each image may contain different number of common targets.

1.3 Problem Formulation

- Image set $I = I_i, i = 1...M$, images may contain different number of targets, goal is to extract all common targets.
- Given I_i generate proposal set $P_i = p_i^k, k = 1...K_i$, set large value for K, to make sure that the object proposal set covers all potential common targets.
- Cos for indefinite number of common targets is transformed into a labeling problem, given p_i^k, $x_i^k = 1$ for foreground, else 0.
- Union set is viewed as final segmentation result

$$R_i = \bigcup \{p_i^k | x_i^k, k \leq K_i\}$$

- Here, the labeling problem in a completely connected network., where each object proposal [OP] is a node, and connected with weighted edges.
- Multiple OP of each image is conducted separately, but closely related to other images of the collection.
- For each image I_i, choose a proposal in every selecting loop to be the real foreground, and in choosing the new loop, we remove the node of the previous proposal to make sure this new proposal would be considered a target, whether this will be chosen as a target depends totally on the labels of other images.

- **Therefore**, segmentation problem of image of I_i finally becomes finding an optimal labeling set $x_i = \{x_i^k | k = 1...K_i; x_k \in \{0, 1\}\}$ by max the energy function (refer to the paper) : function of weights of the edges and some other constraints.
- Weight is non-zero and numerically equal to a similarity score between the proposals (to be introduced later). The constraints mean, for each image only one proposal could be selected per loop, and every proposal in image I_i can be selected only once throughout the selection procedure.
- The formulation is based on the fact that common targets have same characters, and maximizing overall similarity with additional constraint we can make sure newly chosen object proposal is the most similar one to the chosen proposal of the other image. This can be solved via greedy optimization.

1.4 Co-segmentation for Indefinite Number of Common Targets

Key is to adaptively determining the number of targets, which require fully extracting the potential targets and then mining the consistent relationships shared by the common targets.

1.4.1 Overall Framework

1. Category independent OP generated.
2. Connected graph with all proposals as nodes and edge weights as proposal similarities.
3. For reliable similarity, adaptive feature weight selection algorithm.
4. Multiple common targets [MCT] searching, where [MCT] are extracted for each individual image.
5. Terminal condition designed as the common target judging criterion.
6. After termination, simply collect selected proposals.

1.4.2 Object Proposals Generation

1. Very important, directly impacts the performance of Co-segmentation.
2. Measurement of the proposal pool contains mainly two aspects :
   - Diversity : cover as many objects as possible.
   - Representativeness : as few candidates as possible for each object.
3. After a large number of proposals are achieved, a scoring mechanism that combines appearance features and overlap penalty is raised for proposal ranking. There is problem of the proposal containing a local part, but the proposed method could make up for such loss by conducting multiple targets searching.

1.4.3 Weighted Graph Construction

- Usual way : measuring similarity between every two proposals.
- Choosing fixed features for similarity is not a good option. Adopting a flexible and reliable proposal similarity measurement. Here : Unsupervised self-adaptive similarity measurement is introduced for calculating edge weights. Highly efficient and easy to implement. Example in two images colors might be same, in two other images color might be drastically different.
- Use iterative weights setting mechanism for the features. Initial proposal labels using loopy belief algo previously, and then iterating to maximize a function.

- The intuitive intention : encourage selected common targets to be globally consistent while keeping a low variance to make the similarity metric more reasonable and representative.

1.4.4 Common Targets Multi-Search Strategy

Adaptive common target searching strategy that can deal with any numbers of targets.

- For more common targets , remove previously discovered ones from the candidate pool.
- Initialize labels x^* from prev algo. Basically selecting most likely common targets, by removing the prev most likely common target.
- Get an adaptive threshold.

2 Video Object Co-segmentation by Regulated Maximum Weight Cliques

[2]

2.1 Abstract

- Novel approach for object co-segmentation in arbitrary videos by sampling, tracking, and matching [OP] via a Regulated Maximum Weight Clique [RMWC] extraction scheme.
- Achieves good results by pruning away noisy segments in video through selection of [OP] tracklets that are spatially salient and temporally consistent, and by iteratively extracting weighted groupings of objects with similar shape and appearance (with-in and across videos).
- Approach is general and handles : multiple objects, temporary occlusions, objects going in and out of view, also doesn't make any prior assumption on the commonality of the objects in the video collection.
- Keywords : Video Segmentation, Co-segmentation

2.2 Introduction and Related Work

Goal is to discover and segment objects from a video collection in an unsupervised manner.

- Video Co-segmentation is natural extension of Image Co-segmentation.
- In general for video cos, appearance info to group pixels in a spatio-temporal graph and/or employ motion segmentation techniques to separate objects by using motion cues.
- Previous work use strong assumptions of single class of object common to all videos.
- The work has the following advantages :
  - Employs object tracklets as opposed to pixel-level or region-level to perform clustering. The perceptual grouping of pixels before matching reduces segment fragmentation and leads to a simpler matching problem.
  - No approximate solution. [RMWC] has an optimal solution. Using only object tracklets keeps the computation cost low.
  - Can handle occlusions, or objects going in and out of the video because the object tracklets are temporally local and there is no requirements for the object to continuously remain in the field of view of the video. Also no limit on the number of object classes in the each video and number of common object classes in the video collection. Therefore more general.

  - Different from [MWC], in that it is regulated by intra-clique consistency term, as a result produces more global consistency.

2.3 Regulated Maximum Weight Clique based Video Co-segmentation

2.3.1 Framework

Two stages :

1. Object Tracklet Generation : generate [OP] for each frame and use each of them as a starting point and track the object proposals backward and forward throughout the whole video seq, and generate reliable tracklets from the track set and perform non-maxima suppression to remove noisy or overlapping proposals.
2. Multiple Objects Co-segmentation by Regulated Maximum Weight Cliques : Tracklets as node, and the nodes are weighted by tracklet similarity, and edges with weight below a threshold are removed. [RMWC] to find objects ranked by score which is a combination of intra-group consistency and Video Object scores.

2.3.2 Object Tracklets Generation

- Generate a number of [OP]. Each proposal has a Video Object Score : combination of motion and appearance.
$$S^{object}(x) = A(x) + M(x)$$
- $A(x)$: appearance score described directly by algo. High for regions with closed boundary in space, different appearance from its surroundings and is salient.
- $M(x)$: motion score defined as the average frob norm of optical flow gradient around the boundary of object proposal.
- Efficient Object Proposal Tracking :
  - Track every object proposal from each frame backward and forward to form a number of tracks for the object.
  - Combined color (color histograms to model appearance)+ location(overlap ratio) + shape similarity (contour of region in normalized polar coordinates and sampling it from 0 - 360 deg to form a vector) and then dot product for the first and last.
  - Greedy tracking : most similar object proposal is selected to be tracked down, computationally requires finding index of max value in a specific row of the similarity matrix and hence economical.
- Non-maximum Suppression [NMS] for Object Proposal Tracks :
  - Need to prune duplicate (near-duplicate) tracks.
  - Video Object score for one track is obtained, and see $R_{overlap} > 0.5$ and remove them.
  - After [NMS] small percentage of total tracks are retained, and to ensure validity of the track associations, remove associations that are 1.5 std from the mean track similarity.

2.3.3 Multiple Object Co-segmentation by [RMWC]

After object tracklets are obtained, need salient object groupings in the video collection. Grouping problem is formulated as Regulated Maximum Weight Clique.

- Clique Problems :
  - Given G = (V,E,W), a clique is complete subgraph of G, i.e. one whose vertices are pairwise adjacent.

  - Maximal Clique is a complete subgraph not contained in any other complete subgraph.
  - Finding all maximal Clique is NP-hard. Maximum Clique problem is to find the Maximum complete subgraph and Maximum Weight Clique problem deals with finding the Clique with max weight.

- Problem Constraints :
  - Object Proposal Tracklets [OPT] : similar appearance both in video and across video, for in-video L channel used, for across a,b also used.
  - Shape of same object would not change in the same video, and hence used for building tracklets of same objects in a video.
  - Dominant object => high Video Object Score [VOS]
  - Tracklets generated by an object should have low variation.
- Graph Structure :
  - Object tracklets [OT] are nodes, inter and intra video edges created as described above.
  - Weak edges removed by a threshold.
- RMWC :
  - Get weight of node.
  - According to formulation : Clique that has the highest score represents the object with largest combined score of inter-object consistency and objectness. Use NP hard formulation, but doesn't hinder its usage, as number of tracklets are limited, and takes less than a second on standard laptop.

3 Object-Based Multiple Foreground Video Co-Segmentation via Multi-State Selection Graph

[3]

3.1 Abstract

- Multiple foreground Video Co-segmentation for a set of videos.
- Foreground object in each frame considering intra-video coherence of the fg as well as fg consistency among the different videos in the set.
- Multiple foreground handled by multi-state selection graph, node is a video frame, can take multiple labels that correspond to different objects. Also indicator matrix to handle incorrect classification of irrelevant regions.
- Iterative algo to optimize the function.
- Index terms : Video Co-segmentation, Multiple Foregrounds, object-based Segmentation

3.2 Introduction

- Video foreground Co-segmentation aims at jointly extracting the main common objects present in a given set of videos.
- Low level may not accurately discriminate fg and bg. Also object based method for single video do not exploit joint info between the videos. Here : handle Multiple fg object in Multiple videos.

- OP as basic pre-processing. Mid-level features result in more robust and meaningful separation of fg and bg.
- Graph : nodes is video frames, state : to indicate which object proposal is chosen.
- Edges between adjacent frames in a video so as to enforce spatio-temporal smoothness of the trajectory of the fg object, while edges between frames of different videos are added to measure foreground consistency.
- Multi state selection graph [MSG] for multiple states of the nodes to handle multiple Foregrounds. The basic subgraph is replicated multiple times with each replicated subgraph representing a particular foreground object. Can be optimized using existing methods.
- Relax the condition of existence of the common foreground in all the videos, the method will segment unrelated regions in place of the missing object, and use and indicator matrix to correctly deal with the missing common objects.

3.3 Multi-state selection Graph

- Object in a video is a fg if :
  - High appearance contrast relative to the bg.
  - Trajectory of fg object across consecutive video frames is smooth, appearance and shape are also similar across frames.
  - In a video fg object appears in each frame.
  - Additional constraint : on common fg object that they maintain a consistent appearance across different videos.
- Basic subgraph G = (V, E). Define energy function, ψ for nodes, ϕ for edges, and u_n a state taken by each node n in a discrete space. Configuration of states can be done by minimizing the energy function. We seek multiple solutions that are as independent as possible. Introduce a diversity term and define a new optimization problem for MSG.
- Replicate the basic subgraph K-1 times, to get a total of K, diversity term is incorporated as the edges between the corresponding nodes in the basic subgraphs, and combined into a unified graph to get a new energy function and it shares the same formulation as the standard graph, so it can be solved directly by existing energy minimization methods to yield all of the multiple states at once.

3.4 Object Based Video Co-segmentation

Suppose V videos, and each video consists of T_v frames. In the MSG, intra-video edges placed between the nodes of adjacent frames, and inter-video edges are fully connected.

3.4.1 MSG for Video Co-segmentation

- MSG selects same number of states for each node, i.e. the object must appear in each frame of each video. Assumption doesn't hold in general. Use indicator matrix $Y \in \mathbb{R}^{V \times K}$, with $y_{vk} = 1$ to denote video v contains object k.
- Get a new energy function with y_{vk}.

3.4.2 Term Definitions

- Unary Term :

  - $\Psi(.)$ combines three factors (objectness score, motion score, saliency score) for determining the likelihood that an object candidate is the Foreground. Saliency score is considered since the object may not always be moving. Co-saliency is different from saliency : discovers common saliency among multiple images.
  - Here : compute co-saliency map for each frame and then calculate the mean co-saliency value for each candidate region as the saliency score $S(u)$.
- Intra Video Term :
  - $\Phi_a(.,.)$ provides a spatio temporal smoothness constraint between neighboring frames in an individual video.
  - Uses estimated based on iou with wrapped region w.r.t optical flow mapping.
- Inter-video Term :
  - $\Phi_b(.,.)$ measures Foreground consistency among the videos, considering χ^2 color distances of color histograms and HOG descriptors.
- Diversity Term :
  - $\Delta(.,.)$ to avoid selecting the same object in different candidate series. Again IOU.

3.4.3 Optimization Procedure

- In special cases with fixed indicator matrices $\mathbf{Y}$, can be solved directly using existing formulations.
- If not fixed, develop iterative procedure to approximately update two sets of variables (Y, U) until convergence.
- Multiple Foreground video Co-segmentation method :
  - Initialize Y to all 1
  - Solve for u.
  - Update Y. Irrelevant regions have relatively lower unary scores than actual foregrounds. Once y_{vk} updated to 0, can never become 1, to avoid this instead of setting it to 0, set it to $\epsilon(0.0001)$.
  - Termination : if Y doesn't change or upon reaching max number of iterations.
  - Pixel-level Post process : Refine the selected objects through a pixel-level post-process by using a spatiotemporal graph based segmentation method.

4 Convolutional Gated Recurrent Networks for Video Segmentation

[4]

4.1 Abstract

- Novel approach to implicitly utilize temporal data in videos for online Segmentation. Relies on FCN embedded into a gated recurrent architecture.
- The design receives a sequence of consecutive video frames and outputs the segmentation of the last frame.
- Convolutional gated Recurrent networks are used for the recurrent part to preserve spatial connectivities in the image.

- Works on both online and batch segmentation. Tested for both binary and semantic segmentation part.

4.2 Introduction

- **Video segmentation** extensively investigated using classical approaches. Mainly focuses on semi-supervised approaches that propagate the labels in one or more annotated frames to the entire video.
- **Gated Recurrent Architectures** alleviate problem of vanishing or exploding gradients in RNN. LSTM is one of the earliest attempts to design it. Gated Recurrent Unit [GRU] is a more recent attempt, having similar performance to LSTM with reduced number of gates thus fewer parameters.
- Problem : They accept only vectors and hence do not preserve spatio-temporal information. Convolutional GRU can circumvent the problem and has been used for video captioning and action recognition.
- Here : Gated Recurrent FCN. Contributions :
  - Incorporate temporal data to FCN for video Segmentation. Convolutional Gated Recurrent FCN to efficiently utilze spatiotemporal information.
  - End-to-end model for video segmentation.
  - Experimental analysis on binary segmentation and video semantic segmentation.

4.3 Background

Review of FCN and RNN.

4.3.1 Fully Convolutional Networks (FCN)

- All fully connected layers are replaced with convolution layers. Allows input of any size, since it is not restricted to a fixed output size, fully connected layers. Can get a coarse segmentation output (heat map) by only one forward pass of the network.
- Need to upsample the coarse map, and instead of simple bi-linear interpolation, adaptive up-sampling is shown to have better result. Can have learnable layers to learn the up-sampling weights through back-propagation. These are called deconvolution layers.
- Skip architecture can be used for finer Segmentation. Here, heat maps from earlier architecture are merged with the final heat map for an improved Segmentation.

4.3.2 Recurrent Neural Networks

- RNN can be applied on a sequence of inputs and are able to capture the temporal relation betwen them.
- Hidden unit in each recurrent cell allows it to have dynamic memory that is changing according to what it had before.
- For longer vectors it causes vanishing gradients. ated recurrent architectures have been proposed as a solution and empiricially useful for many tasks.
- **Long Short Term Memory (LSTM)** LSTM utilizes three gates to control flow of signal : input, output, forget gate, each with own set of weights and learned with back-propagation. At the inference stage, values in the hidden unit ican be roughly interpreted as a memory, and used for prediciton of the current state.

- **Gated Recurrent Unit (GRU)** Same principal as LSTM, with simpler architecture, less computationally expensive, and requires less memory.

4.4 Method

A Recurrent Fully Convolutional Network (RFCN) is designed that utilizes tge spatio-temporal information. Two approaches are explored : conventional recurrent units, convolutional recurrent units.

4.4.1 Conventional Recurrent Architecture for Segmentation

- **RFC-LeNet**
  - Fully convolutional version of LeNet which is a shallow network, used for baseline comparisons.
  - Output of deconvolution fo 2D map of dense predicitons is flattened into 1D vector as the input to a conventional Recurrent unit, and the unit takes this vector for each frame in teh sliding window and outputs the Segmentation of the last frame.
- **RFC-12s**
  - Apply the recurrent layer on the down-sampled heatmap before deconvolution.
  - The recurrent unit operates on coars maps to produce a coarse map corresponding to the last frame in the sequence.

4.4.2 Convolutional Gated Recurrent Architecture (Conv-GRU) for Segmentation

- Conventinal recurrent units are designed for text processing and not image processing, and directly using for images has pains :
  - The size of weight parameters becomes very large since vectorized images are large
  - Spatial connectivity between pixels are ignored.
- Convolutional Recurrent units, convolve 3D weights with their input. Dot products replaced by convolutions. Learning filters that convolve with the entire imagee instead of individual weights for pixels, makes it much more efficient.
- **RFC-VGG** Intermediate faeture maps are fed into a convlutional atead recurrent unit, and then a convolutional layer converts its output to a heat map.
- **RFCN-8s** : Recurrent verison of FCN-8s architecture.

5 Structural RNN : Deep Learning on Spatio-temporal Graphs

[5]

5.1 Abstract

- Deep RNN capable at modelig seqeunces, lack an intuitive highlevel spatio-temporal sturcutre which are popular tool for imposting such highlevel intuitions in the formulations of real world problems.
- Develop a scalable method for casting an arbitratry spatio-temporal graph as a rich RNN mixture that is feedforward fully differentialble, and jointly trainable.

- Shows imporvement over sotawith a large margin.

5.2 Introduction

- Spatio-temporal [st] graphs are popular tool for representing such high-level spatio-temporal structures. The nodes represent the Problem components, and the edges capture their spatio-temporal interactions.
- Here [st] graph into sturctural RNN [S-RNN].
- In high-level steps, given any arbitrary st-graph :
  - Roll it out in time and decompose it into a set of contributing factor components.
  - The factors identify the independent components that collectively determine one decision and are derived from both edges and nodes fo the st-graph.
  - Semantically group thefactor componeonts and represent each group using one RNN which results in the desired RNN mixture.
  - Main challenges of the transformation is :
    - Making the RNN mixture as rich as possible to enable learning complex functions : represent each st factor using one RNN.
    - Keeping the RNN mixture scalable with respect to size of the input st-graph : factor sharing between RNNs.
- Proposed method is generic and applicable to any problem formulated using st-graph. Enjoys underlying high-level structure.

5.3 SRNN architecture

5.3.1 Representation of spatio-temporal Graphs

- St-graph : $G = (V, E_s, E_T)$ whose structure (V, E_s) rolls over time through edges E_T.
- The nodes $v \in V$ and $e \in E_s \cap E_T$ of the graph repeats over time. In the unrolled st-graph, the ndoes at a given time step t are connected with undirected spatio-temporal edge $e = (u, v) \in E_s$ and the nodes at adjacent time steps (node u at time t and v at time $t+1$) are connected with undirected temporal edge iff $(u, v) \in E_T$.
- Given a st-graph and the feature vectors associated with the nodes x_v^t and edges x_e^t, the goal is to predict the node labels (or real value vectors) y_v^t at each time step t. Label y_v^t is afffectedby both its node and its interactions with other nodes (edges), leading to an overall complex system.
- Here : derive SRNN structure from the factor-graph representation of st-graph. The factor graph has a factor function $\Psi_v(y_v, x_v)$ for each node and a pairwise factor $\Psi_e(y_{e(1)}, y_{e(2)}, x_e)$ for each edge.
- Sharing factors between nodes : Each factor has parameters that needs to be learned. Semantically similar nodes can optionally share factors.
- Partition the nodes as $C_v = \{V_1, ..., V_P\}$ where V_P is a set of semantically similar nodes and they all use the same node factor Ψ_{V_p}. Similaryly for edges. For modeling flexibility the edge factors are note shared across the edges in E_s and E_T.
- To predict teh label of the node $v \in V_P$: consider node factor, edge factors connected to v in the factor graph. Formally the node factor and edge factor are neighbors if there exists a node v such taht it connects to both in the factor graph.

5.3.2 Structural RNN from spatio-temporal graphs

- Represent each factor with an RNN. As such there are nodeRNNs R_{V_p} and edgeRNNs R_{E_m} and they are connected iff they are neighbors in the original st-graph, outputs a bipartite graph $G = (R_{E_m}, R_{V_p}, E_R)$.
- The predictions of nodeRNNs interact through edgeRNNs.

5.3.3 Training structural-RNN architecture

- The nodeRNN at each time step concatenates the node feature x_v^t and the outsputs of edgeRNNs it is connected to and predictst hte node label. At the time of training the errors int eh prediction are back propagated through the nodeRNN and edge RN involved during the forward pass.
- Parameter sharing and strucutred feature space

References

[1] K. Li, J. Zhang, and W. Tao, "Unsupervised co-segmentation for indefinite number of common foreground objects," *IEEE Transactions on Image Processing*, vol. 25, pp. 1898–1909, April 2016.

[2] D. Zhang, O. Javed, and M. Shah, *Video Object Co-segmentation by Regulated Maximum Weight Cliques*, pp. 551–566. Cham: Springer International Publishing, 2014.

[3] H. Fu, D. Xu, B. Zhang, S. Lin, and R. K. Ward, "Object-based multiple foreground video co-segmentation via multi-state selection graph," *IEEE Transactions on Image Processing*, vol. 24, pp. 3415–3424, Nov 2015.

[4] M. Siam, S. Valipour, M. Jägersand, and N. Ray, "Convolutional gated recurrent networks for video segmentation," *CoRR*, vol. abs/1611.05435, 2016.

[5] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," *CoRR*, vol. abs/1511.05298, 2015.